PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

AF

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	09/765,916
Filing Date	January 18, 2001
First Named Inventor	Frederic Canut et al.
Art Unit	2193
Examiner Name	Insun Kang
Attorney Docket Number	260/087; CA7017922001

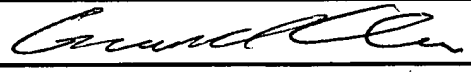
ENCLOSURES ***(check all that apply)***

- ☒ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Reply (pgs).
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR1.52 or 1.53
- ☐ Formal Drawing(s) (sheets)
- ☐ Licensing-related Papers
- ☐ Petition to Correct Inventorship
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) ____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☒ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) *(please identify below)*:

Return Postcard

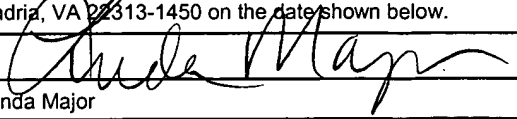
Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm	Bingham McCutchen LLP		
Signature			
Printed Name	Gerald Chan		
Date	March 29, 2007	Reg. No.	51,541

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Signature			
Typed or printed name	Linda Major	Date	March 29, 2007

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:	Group Art Unit: 2193
Frederic Canut et al.	Examiner: Insun Kang
Serial No.: 09/765,916	Confirmation No.: 8270
Filed: January 18, 2001	
For: SYSTEM AND METHOD FOR SOFTWARE CODE OPTIMIZATION	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicants submit this Appeal Brief pursuant to the Notice of Appeal filed on January 29, 2007. Thus, this Appeal Brief is timely filed.

This Appeal Brief is organized as follows per 37 C.F.R. §41.37(c)(1):

I. REAL PARTY IN INTEREST

The real party in interest is **Cadence Design Systems, Inc.** of San Jose, California and is the assignee of this application.

II. RELATED APPEALS AND INTERFERENCES

To the best of Applicants' knowledge, there are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-26 are pending. Claims 1-26 are rejected, and are appealed.

IV. STATUS OF AMENDMENTS

No amendments have been filed after the final rejection dated September 1, 2006.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present section of the Appeal Brief is set forth merely to comply with the requirements of 37 C.F.R. §41.37(c)(v) and is not intended to limit the pending claims in any way.

Independent claim 1 recites:

> A method of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language , the method comprising the steps of:
> (a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program , and comparing the first performance profile with the performance objectives;
> (b) based on the results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language; and
> (c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program.

Independent claim 14 recites:

> A computer-readable medium comprising a sequence of instructions which, when executed by a processor, causes the processor to execute a method for optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:
> (a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;
> (b) based on the results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language; and
> (c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program.

Embodiments of the subject matter of claims 1 and 14 are described in figure 1 and page 6, line 3 to page 7, line 18 of the specification. In particular, figure 1 and page 6, lines 3-7 of the specification describe an embodiment of a method of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language. Page 6, lines 7-8 of the specification describe an embodiment of optimizing a software program such that a resulting first optimized form of the software program is completely independent of the target processor. Page 6, lines 12-13 of the specification also describe an embodiment in which the first optimized form of the software program is at least partially coded in high-level language, and an embodiment of determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives. Page 6, line 16 to page 7, line 6 of the specification describe an embodiment of optimizing the first optimized form of the software program, based on the results of comparing the first performance profile with the performance objectives, such that a resulting second

optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language, if the performance objectives are not met by the first optimized form of the software program. Page 7, lines 6-18 of the specification describe an embodiment of flagging a portion to indicate that the portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program. Page 9, line 3 to page 11, line 2 of the specification describe an embodiment of using a computer-readable medium to implement a method for optimizing a software program for a target processor to meet performance objectives.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 1-3, 8-16 and 21-26 are patentable under 35 U.S.C. §103(a) over U.S. Patent Application Publication No. US 2003/0005419 Pieper et al. ("Pieper") in view of "Portable Software Library Optimization", 2/1998, Cain et al. ("Cain").

B. Whether claims 4-7 and 17-20 are patentable under 35 U.S.C. §103(a) over Pieper in view of Cain and IEEE 0-7803-5041-3/99, 1999, Kum et al. ("Kum").

VII. ARGUMENT

A. Claim Rejections under 35 U.S.C. §103

Claims 1-3, 8-16 and 21-26

Applicants respectfully submit that claims 1-3, 8-16 and 21-26 are patentable over Pieper in view of Cain because the combination of these references does not disclose or suggest each and every limitation recited in these claims.

Present independent claim 1 recites optimizing a software program such that *a resulting first optimized form of the software program is completely independent of the target processor* and is at least partially coded in the high-level language. (Emphasis Added) Pieper does not disclose or suggest the above limitations. Rather, Pieper discloses that an optimized code 60 output by the optimization processes 58 is in an intermediate level program code language that is substantially independent of the architecture of the target processor (figure 2 and paragraph 30). The term "substantially independent" in Pieper inherently requires that a portion or part of the optimized code of Pieper be dependent or not independent of the architecture of the target processor. In other words, since Pieper specifically discloses that the first optimized code 60 is substantially independent of the architecture of target processor 12, Pieper actually requires that a portion or part of the first optimized code 60 is not completely independent. This assertion is conceded in the Office Action, dated March 6, 2006, in page 8, section 7, in which the Examiner stated, "Pieper uses the term, 'substantially independent' which indicates that some portion is not independent." Applicants further note that the Examiner has admitted that the term "substantially" and "completely" convey different scopes (page 9 of Office Action). As such, it would be inappropriate to consider the term "substantially" in Pieper to be anticipatory of the limitation "completely." For at least the foregoing reasons, Applicants respectfully submit that Pieper does not disclose or suggest a first optimized form of the software program that is *completely independent* of the target processor, as recited in claim 1.

According to the Office Action, paragraph 20 on page 3 of Pieper discloses "machine-independent code," which the Examiner considers to be a resulting first optimized form of the software program that is completely independent of the target processor. However, as the Office

Action has admitted, in Pieper, the machine-independent code is used during a "compilation process," and therefore, the machine-independent code is certainly not a result of an optimization process. This is further evidenced by the disclosure of Pieper, which distinguishes a compilation from an optimization process. In particular, figure 2 and paragraph 18 of Pieper disclose that a first set of computer program instructions in a high level program instruction language 52 is converted (compiled) by compilation processes 50 into a second set of computer program instructions in a low level program instruction language 74. This operation by Pieper is merely a compilation and is not an optimization. Figure 2 and paragraph 20 of Pieper also disclose that, during compilation process 54, prefetch instructions are inserted into the machine-independent code 52 to generate an intermediate form 56. According to Pieper, this prefetch instruction insertion occurs before the optimization 58. As such, the machine-independent code as described in Pieper clearly cannot be considered "a resulting first optimized form of the software program" as recited in claim 1.

Cain also does not disclose or suggest the above limitations, and therefore, does not remedy the deficiencies of Pieper. Applicants further note that Cain is merely relied on to reject subject matter of present independent claim 1 related to, "flagging the at least one portion to indicate that the at least one portion is dependent on the target processor" and is not relied upon for the disclosure of the above discussed limitations. Since neither Pieper nor Cain discloses or suggests the above claim limitations, they cannot be combined to form the resulting subject matter of claim 1.

Because Pieper and Cain do not disclose or suggest a resulting first optimized form of the software program that is completely independent of the target processor, these references also do not disclose or suggest determining a first performance profile for such first optimized form of the software program, and comparing such first performance profile with the performance objectives, as recited in claim 1.

For at least the foregoing reasons, Applicants respectfully submit that present independent claim 1 is patentable over Pieper, Cain, and their combination.

For at least the same foregoing reasons, Applicants respectfully submit that present independent claim 14 is patentable over Pieper , Cain, and their combination because present independent claim 14 recites similar limitations as those of claim 1.

Applicants respectfully submit that the remaining claims 2-3, 8-13, 15-16 and 21-26 are patentable over Pieper, Cain, and their combination due to their respective dependence on independent claims 1 and 14.

B. Claim Rejections under 35 U.S.C. § 103

Claims 4-7 and 17-20

Applicants respectfully submit that claims 4-7 and 17-20 are patentable over Pieper in view of Cain and Kum under 35 U.S.C. §103 due to their respective dependence on present independent claims 1 and 14 for at least the reasons discussed above in section VII. A.

CONCLUSION

For the above reasons, Applicants respectfully request that the Board of Patent Appeals and Interferences overrule the Examiner and allow claims 1-26.

The Commissioner is authorized to charge any fees due in connection with the filing of this document to Bingham McCutchen's Deposit Account No. **50-2518,** referencing billing number **7017922001**. The Commissioner is authorized to credit any overpayment or to charge any underpayment to Bingham McCutchen's Deposit Account No. **50-2518,** referencing billing number **7017922001**.

Respectfully submitted,
Bingham McCutchen LLP

Date: March 29, 2007

By: 
Gerald Chan
Registration No. 51,541

Three Embarcadero Center
San Francisco, CA 94111
Telephone: (650) 849-4960
Facsimile: (650) 849-4800

VIII. APPENDIX A: CLAIMS APPENDIX

Listing of Appealed Claims 1-26

1 A method of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;

(b) based on the results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language; and

(c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program.

2. The method of claim 1, further comprising steps of:

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

3. The method of claim 1, further comprising:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and includes portions coded in a low-level language of the target processor.

4. The method of claim 1 in which step (a) comprises the act of deriving a floating point implementation.

5. The method of claim 1 in which step (a) comprises the act of deriving a fixed point implementation.

6. The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

7. The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

8. The method of claim 1 in which step (a) comprises the act of implementing reference code.

9. The method of claim 8 in which the act of implementing reference code comprises code profiling.

10. The method of claim 1 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.

11. The method of claim 1 in which step (b) comprises the act of tuning low-level functions.

12. The method of claim 3 in which step (d) comprises the act of manual assembly optimization.

13. The method of claim 1 in which step (b) comprises the act of feature tuning.

14. A computer-readable medium comprising a sequence of instructions which, when executed by a processor, causes the processor to execute a method for optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;

(b) based on the results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language; and

(c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program.

15. The computer-readable medium of claim 14, in which the method further comprises the steps of:

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

16. The computer-readable medium of claim 14, wherein the method further comprises:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and is includes portions coded in a low-level language of the target processor.

17. The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a floating point implementation.

18. The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a fixed point implementation.

19. The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

20. The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

21. The computer-readable medium of claim 14 in which step (a) comprises the act of implementing reference code.

22. The computer-readable medium of claim 21 in which the act of implementing reference code comprises code profiling.

23. The computer-readable medium of claim 14 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.

24. The computer-readable medium of claim 14 in which step (b) comprises the act of tuning low-level functions.

25. The computer-readable medium of claim 16 in which step (d) comprises the act of manual assembly optimization.

26. The computer-readable medium of claim 14 in which step (b) comprises the act of feature tuning.

IX. APPENDIX B: EVIDENCE APPENDIX

None.

X. APPENDIX C: RELATED PROCEEDINGS APPENDIX

None.